RELATIONAL DATABASE DESIGN

CSC 350

How do we design a database schema? How do we design a *good* database schema? How do we design a *normalized* database schema? classroom(building, <u>room_number</u>, capacity) department(dept_name, building, budget) course(<u>course_id</u>, title, dept_name, credits) instructor(<u>ID</u>, name, dept_name, salary) section(<u>course_id</u>, <u>sec_id</u>, <u>semester</u>, <u>year</u>, building, room_number, time_slot_id) teaches(<u>ID</u>, <u>course_id</u>, <u>sec_id</u>, <u>semester</u>, <u>year</u>) student(<u>ID</u>, name, dept_name, tot_cred) takes(<u>ID</u>, <u>course_id</u>, <u>sec_id</u>, <u>semester</u>, <u>year</u>, grade) advisor(<u>s_ID</u>, i_ID) time_slot(<u>time_slot_id</u>, <u>day</u>, <u>start_time</u>, end_time) prereq(<u>course_id</u>, prereq_id)

Can we simplify this schema?

SCHEMA DESIGN

classroom(building, room_number, capacity)
department(dept_name, building, budget)
course(course_id, title, dept_name, credits)
instructor(ID, name, dept_name, salary)
section(course_id, sec_id, semester, year, building, room_number, time_slot_id)
teaches(ID, course_id, sec_id, semester, year)
student(ID, name, dept_name, tot_cred)
takes(ID, course_id, sec_id, semester, year, grade)
advisor(s_ID, i_ID)
time_slot(time_slot_id, day, start_time, end_time)
prereq(course_id, prereq_id)

ID	name	salary	dept_name	building	budget
22222	Einstein	95000	Physics	Watson	70000
12121	Wu	90000	Finance	Painter	120000
32343	El Said	60000	History	Painter	50000
45565	Katz	75000	Comp. Sci.	Taylor	100000
98345	Kim	80000	Elec. Eng.	Taylor	85000
76766	Crick	72000	Biology	Watson	90000
10101	Srinivasan	65000	Comp. Sci.	Taylor	100000
58583	Califieri	62000	History	Painter	50000
83821	Brandt	92000	Comp. Sci.	Taylor	100000
15151	Mozart	40000	Music	Packard	80000
33456	Gold	87000	Physics	Watson	70000
76543	Singh	80000	Finance	Painter	120000

The new inst_dept table...good idea or bad idea?

FUNCTIONAL DEPENDENCIES

A relationship where one attribute uniquely determines another attribute.

x "uniquely determines" y $x \rightarrow y$

ID	name	salary	dept_name	building	budget
22222	Einstein	95000	Physics	Watson	70000
12121	Wu	90000	Finance	Painter	120000
32343	El Said	60000	History	Painter	50000
45565	Katz	75000	Comp. Sci.	Taylor	100000
98345	Kim	80000	Elec. Eng.	Taylor	85000
76766	Crick	72000	Biology	Watson	90000
10101	Srinivasan	65000	Comp. Sci.	Taylor	100000
58583	Califieri	62000	History	Painter	50000
83821	Brandt	92000	Comp. Sci.	Taylor	100000
15151	Mozart	40000	Music	Packard	80000
33456	Gold	87000	Physics	Watson	70000
76543	Singh	80000	Finance	Painter	120000

department → budget

Functional Dependencies show us where we need to "decompose" a relation into smaller relations to avoid redundancy.

FUNCTIONAL DEPENDENCIES

Not all decompositions are good.



city

salary

75000

67000

If R = (a, b, c)

 $R_1 = (a, b)$ and $R_2 = (b, c)$ are lossless decompositions of R if:

R₁ and R₂ are lossless decompositions of R if:

 $R_1 \bowtie R_2 = R$

The domain of an attribute is the set of all possible values it can take.

int x;

(x's domain is { -2,147,483,648..., 0, ...2,147,483,647 }

department d ;
(d's domain is { math, computer science, biology, ... }

A domain is considered atomic if it is made of indivisible units.

Which of these are atomic?

name = { "bob jones", "steve smith", "suzy walters", ... }

id = { 12, 55, 233, ... }

course_number = { cs101, cs105, bio220, bio320, ... }

A relation is considered to be in first normal form (1NF) if:

- The domain of each attribute is atomic.
- The value of each attribute contains only a single value from that domain.
- Each tuple contains a unique key.

Customer					
Customer ID	First Name	Surname	Telephone Number		
123	Pooja	Singh	555-861-2025, 192-122-1111		
456	San	Zhang	(555) 403-1659 Ext. 53; 182-929-2929		
789	John	Doe	555-808-9633		

Customer						
Customer ID	First Name	Surname	Telephone Number1	Telephone Number2		
123	Pooja	Singh	555-861-2025	192-122-1111		
456	San	Zhang	(555) 403-1659 Ext. 53	182-929-2929		
789	John	Doe	555-808-9633			

FIRST NORMAL FORM (1NF)

Customer						
Customer ID First Name Surname Telephone Number						
123	Pooja	Singh	555-861-2025			
123	Pooja	Singh	192-122-1111			
456	San	Zhang	182-929-2929			
456	San	Zhang	(555) 403-1659 Ext. 53			
789	John	Doe	555-808-9633			

Cus	stomer Name		Customer Telephone Number		
Customer ID	First Name	Surname	Customer ID	Telephone Number	
123	Pooja	Singh	123	555-861-2025	
456	San	Zhang	123	192-122-1111	
789	John	Doe	456	(555) 403-1659 Ext. 53	
	·	·,	456	182-929-2929	
			789	555-808-9633	

A relation is considered to be in second normal form (2NF) if:

- The relation is in first normal form.
- No attribute depends on only part of the primary key.

Electric Toothbrush Models						
Manufacturer	Model	Model Full Name	Manufacturer Country			
Forte	X-Prime	Forte X-Prime	Italy			
Forte	Ultraclean	Forte Ultraclean	Italy			
Dent-o-Fresh	EZbrush	Dent-o-Fresh EZbrush	USA			
Kobayashi	ST-60	Kobayashi ST-60	Japan			
Hoch	Toothmaster	Hoch Toothmaster	Germany			
Hoch	X-Prime	Hoch X-Prime	Germany			

Is this relation in second normal form?

SECOND NORMAL FORM (2NF)

Electric Toothbrush Manufacturers

Manufacturer	Manufacturer Country
Forte	Italy
Dent-o-Fresh	USA
Kobayashi	Japan
Hoch	Germany

Electric Toothbrush Models

Manufacturer	Model	Model Full Name
Forte	X-Prime	Forte X-Prime
Forte	Ultraclean	Forte Ultraclean
Dent-o-Fresh	EZbrush	Dent-o-Fresh EZbrush
Kobayashi	ST-60	Kobayashi ST-60
Hoch	Toothmaster	Hoch Toothmaster
Hoch	X-Prime	Hoch X-Prime

Is this relation in second normal form?

A relation is considered to be in third normal form (3NF) if:

- The relation is in second normal form.
- No attribute describes anything except facts about the primary key.

"[Every] non-key [attribute] must provide a fact about the key, the whole key, and nothing but the key."

- A Simple Guide to Five Normal Forms in Relational Database Theory

Tournament Winners

Tournament	<u>Year</u>	Winner	Winner Date of Birth
Indiana Invitational	1998	Al Fredrickson	21 July 1975
Cleveland Open	1999	Bob Albertson	28 September 1968
Des Moines Masters	1999	Al Fredrickson	21 July 1975
Indiana Invitational	1999	Chip Masterson	14 March 1977

Winner Dates of Birth

Tournament	<u>Year</u>	Winner	<u>Winner</u>	Date of Birth
Indiana Invitational	1998	Al Fredrickson	Chip Masterson	14 March 1977
Cleveland Open	1999	Bob Albertson	Al Fredrickson	21 July 1975
Des Moines Masters	1999	Al Fredrickson	Bob Albertson	28 September 1968
Indiana Invitational	1999	Chip Masterson		·

Sometimes it is necessary to *denormalize* from 3NF for performance reasons.



What query would give you the YTD sales for customer 123?

SELECT SUM(p.price) as YTDSales FROM Customers c, Invoices i, InvoiceDetails id, Products p WHERE c.CustomerId = i.CustomerId AND i.InvoiceId = id.InvoiceId AND id.ProductId = p.ProduceId AND c.CustomerId = 123 AND i.InvoiceDate BETWEEN DATE_TRUNC('year', NOW()) and NOW();

What query would give you the YTD sales for customer 123?



SELECT YTDSales FROM Customers WHERE CustomerId = 123;

What's required to maintain data integrity in this situation?

Sometimes it is necessary to *denormalize* from 3NF for performance reasons.

When to denormalize?

https://stackoverflow.com/questions/4301089/when-to-denormalize-a-database-design